We claim:

1. A method comprising:

mapping a plurality of physically non-contiguous sections of memory into a logically contiguous section of memory; and,

for each computing unit of a plurality of computing units, allocating a portion of the logically contiguous section of memory addressable by a pointer plus a static offset corresponding to the computing unit,

wherein the static offset for each computing unit is equal to a static offset initially determined at initial allocation of memory for the plurality of computing units.

2. The method of claim 1, wherein the portion of the logically contiguous section of memory allocated for each computing unit includes memory local to the computing unit.

3. The method of claim 1, further comprising:

dynamically passing out the portion of the logically contiguous section of memory to each computing unit of the plurality of computing units as the computing units need additional memory;

upon the logically contiguous section of memory being completely passed out to the plurality of computing units,

mapping a second plurality of physically non-contiguous sections of memory into a second logically contiguous section of memory;

for each computing unit of the plurality of computing units, allocation a portion of the second logically contiguous section of memory addressable by a

pointer plus the static offset corresponding to the computing unit; and,

dynamically passing out the portion of the second logically contiguous section of memory to each computing unit of the plurality of computing units as the computing units needs additional memory.

4. The method of claim 1, further comprising determining the static offset for each computing unit as equal to the static offset initially determined at the initial allocation of the memory for the plurality of computing units.

5. The method of claim 1, further comprising at the initial allocation of the memory for the plurality of computing units:

determining the static offset for each computing unit of the plurality of computing units; and,

for each computing unit of the plurality of computing units, allocating a portion of memory addressable by a pointer plus the static offset corresponding to the computing unit.

6. The method of claim 5, further comprising dynamically passing out the portion of the memory to each computing unit of the plurality of computing units as the computing units need additional memory.

7. The method of claim 1, wherein the computing unit is one of a computing node and a processor.

8. A method comprising:

determining a static offset for each computing unit of a plurality of computing units;

for each computing unit of the plurality of computing units, allocating a portion of predetermined memory addressable by a pointer plus the static offset corresponding to the computing unit;

dynamically passing out the portion of the predetermined memory to each computing unit of the plurality of computing units as the computing units need additional memory;

upon the predetermined memory being completely passed out to the plurality of computing units,

mapping a plurality of physically non-contiguous sections of additional memory into a logically contiguous section of memory;

for each computing unit of the plurality of computing units, allocating a portion of the logically contiguous section of memory addressable by a pointer plus the static offset corresponding to the computing unit, where the portion of the logically contiguous section of memory is local to the computing unit; and,

dynamically passing out the portion of the logically contiguous section of memory to each computing unit of the plurality of computing units as the computing units need additional memory.

9. The method of claim 8, further comprising, upon the logically contiguous section of memory being completely passed out to the plurality of computing units,

mapping a second plurality of physically non-contiguous sections of memory into

a second logically contiguous section of memory;

for each computing unit of the plurality of computing units, allocation a portion of the second logically contiguous section of memory addressable by a pointer plus the static offset corresponding to the computing unit, wherein the portion of the second logically contiguous section of memory is local to the computing unit; and,

dynamically passing out the portion of the second logically contiguous section of memory to each computing unit of the plurality of computing units as the computing units needs additional memory.

10. A system comprising:

a plurality of computing nodes;

memory shared by the plurality of computing nodes; and,

an allocating mechanism to map a plurality of physically non-contiguous sections of memory into a logically contiguous section of memory and to allocate a portion of the logically contiguous section of memory to each computing node,

wherein the portion of the logically contiguous section of memory is addressable by a pointer plus a static offset corresponding to the computing node and equal to a static offset initially determined for allocating a portion of memory to each computing node.

11. The system of claim 10, wherein the portion of the logically contiguous section of memory allocated to each computing node includes memory local to the computing node.

12. The system of claim 10, wherein each of the plurality of computing nodes comprises a single processor.

13. The system of claim 10, wherein each of the plurality of computing nodes comprises a plurality of processors.

14. A system comprising:

a plurality of computing nodes;

memory shared by the plurality of computing nodes; and,

means for mapping a plurality of physically non-contiguous sections of memory into a logically contiguous section of memory and for allocating a portion of the logically contiguous section of memory to each computing node,

wherein the portion of the logically contiguous section of memory is addressable by a pointer plus a static offset corresponding to the computing node and equal to a static offset for allocating a portion of memory to each computing node.

15. A computing node comprising:

a plurality of processors;

memory shared by the plurality of processors; and,

an allocating mechanism to map a plurality of physically non-contiguous sections of memory into a logically contiguous section of memory and to allocate a portion of the logically contiguous section of memory to each processor,

wherein the portion of the logically contiguous section of memory is addressable by a pointer plus a static offset corresponding to the processor and equal to a static offset initially determined for allocating a portion of memory to each processor.

16. The computing node of claim 15, wherein the portion of the logically contiguous section of memory allocated to each processor is local to the processor.

17. A computing node comprising:

a plurality of processors;

memory shared by the plurality of processors; and,

means for mapping a plurality of physically non-contiguous sections of memory into a logically contiguous section of memory and for allocating a portion of the logically contiguous section of memory to each processor,

wherein the portion of the logically contiguous section of memory is addressable by a pointer plus a static offset corresponding to the processor and equal to a static offset for allocating a portion of memory to each processor.

18. An article of manufacture comprising:

a computer-readable medium; and,

means in the medium for mapping a plurality of physically non-contiguous sections of memory into a logically contiguous section of memory and for allocating a portion of the logically contiguous section of memory to each computing unit of a plurality of computing units,

wherein the portion of the logically contiguous section of memory is addressable by a pointer plus a static offset corresponding to the computing unit and equal to a static offset initially determined at boot time of the plurality of computing units for allocating a portion of memory to each computing unit.